AMENDMENTS TO THE CLAIMS

Please amend Claims 1 and 13, as follows.

Please cancel, without prejudice, Claims 14-15.

Please add new Claims 16-17.

Claim 1. (Currently amended) A process for automatically detecting exceptions in a sequence of pipelined instructions comprising an instruction at which correct state is known and floating point instructions, said process comprising the steps of:

a) automatically inserting a command that tests for ~~raised~~ generated floating point status exceptions into a sequence of instructions to be executed,

b) responding to a test by said command that indicates a generated floating point status exception ~~an exception raised~~ during pipelined execution of the sequence of instructions by returning execution to an instruction in the sequence of instructions at which correct state is known, and

c) executing each instruction in the sequence singly to completion until ~~the~~ a floating point exception is detected ~~again raised~~.

Claim 2. (Original) A process as claimed in Claim 1 in which the command is inserted in the sequence after a last floating point instruction and before floating point status is saved.

Claim 3. (Original) A process as claimed in Claim 2 in which the command is inserted after a branch in the sequence.

Claim 4. (Original) A process as claimed in Claim 2 in which the command stalls the pipeline if the last floating point instruction has not completed execution when status is to be saved.

Claim 5. (Original) A process as claimed in Claim 2 in which the command does not stall the pipeline if the last floating point instruction has not completed execution when status is to be saved.

Claim 6. (Original) A process as claimed in Claim 5 in which floating point status saved is floating point status existing when integer status is saved.

Claim 7. (Original) A process as claimed in Claim 5 in which floating point status saved is floating point status generated by floating point operations which have completed when integer status is saved.

Claim 8. (Original) A process as claimed in Claim 1 in which the command compares accumulated condition of exception status detected during execution of the sequence of instructions with armed floating point exception conditions.

Claim 9. (Original) A process as claimed in Claim 8 in which the command executes and compares accumulated condition of exception status detected when integer status is saved.

Claim 10. (Original) A process as claimed in Claim 8 in which the command raises an exception only if newly accrued exceptions have not previously been committed.

Claim 11. (Original) A process as claimed in Claim 8 in which exception status detected includes exceptions generated by a command for manipulating memory operands used in floating point stack operations.

Claim 12. (Original) A process as claimed in Claim 11 in which no exception is raised if the corresponding exceptions generated by a command for manipulating memory operands used in floating point stack operations are not armed and have already been reported.

Claim 13. (Currently amended) An Apparatus for automatically detecting exceptions in pipelined instructions comprising an instruction at which correct state is known and floating point instructions, said apparatus comprising:

a computer-executable software process which automatically inserts commands that test for ~~raised exceptions indicating~~ generated floating point status exceptions into a sequence of instructions to be executed during dynamic translation of target instructions,

a computer-executable software process for responding to ~~exceptions~~ ones of the commands indicating a generated floating point status exception by rolling execution of a sequence of instructions back to a point at which correct state is known, and

a computer-executable software process for executing each instruction in the sequence singly to completion until ~~the~~ a floating point exception is detected ~~again raised~~.

Claims 14-15. (Cancelled)

Claim 16. (New) A method for automatically detecting floating point exceptions in a sequence of host instructions translated from a sequence of target instructions, said method comprising:

a) automatically inserting a command that detects report of a floating point exception in a sequence of pipelined host instructions translated from a sequence of target instructions;

b) detecting, by the command, a report of a floating point instruction exception during execution of the sequence of pipelined host instructions;

c) executing singly host instructions translated from said sequence of target instructions in response to said command detecting said report in said b), said executing singly starting at a point in the sequence of target instructions at which correct state is known; and

d) detecting a floating point instruction exception during said executing singly host instructions.

Claim 17. (New) The method of claim 16, wherein the inserted command compares floating point status bits at completion of a last floating point instruction with an arming condition of floating point status bits before a commit command in the sequence of pipelined host instructions is executed.